

```

1 Discussion : 2020-07-20
2 -----
3 Saurab
4 -----
5     1.
6     2. gram indices : abcu abc -> 450, bcu -> 679
7     vector dimension: 1 5000
8
9     grams [abc, bcu, cus, ab, bc, cu, us]
10    to avoid iterating from 1 to 5000.
11
12 Shankar
13 -----
14     1. Index Map :
15     2. Completed without query search in relevant words.
16     3. How to implement with relevant words ?
17
18     vector of entry => n = 15
19     no. of words => 1000 = n + k
20
21     (n + k) * n for each wrong word.
22      $n^2 + nk \Rightarrow O(n^2)$ 
23
24     Inverted index =>
25         g1 -> [w1, w203, w500]
26         g2 -> [w1, w609, w999]
27
28
29     During Search:
30     wrong_word wk = [g100, g2, g890, g350]
31
32     wk -> map with inverted_index -> List of List
33     filter distinct -> list of words (relevant)
34
35 -----
36 What is log function ?
37      $2^5 = 32$ 
38      $\log_2(32) = 5$ 
39
40 -----
41
42 Minu
43 -----
44     1. flow completed.
45     2.
46
47 w1: apple => [ap, pp, pl, le, app, ppl, ple]
48 w2:
49 w3:
50
51 w1000:
52
53 First transformation.
54 flatten with value,
55 apple -> ap
56 apple -> pp
57 apple -> pl
58 apple -> le
59 apple -> app
60 apple -> ppl
61 apple -> ple
62 .
63 .
64 .

```

```

65
66 Transformation group by value,
67 ap -> [(apple -> ap), (appliation -> ap), ...]
68 pp -> [()], [()]
69
70 .
71 .
72 .
73
74 map apply to values tuple
75 ap -> [apple, applicaton, . . .], distinct.
76
77 This is inverted index.
78 idf -> N/count values. [Normalization.]
79
80 [1, 0, 5, 6, 9, 0, 0, 8]
81 [0, 0, 0,          0, 100]
82 [0, 0, 0,  1, 1, 0, 0, 0]
83
84 -----
85 n -> input -> f(n) -> ? to compute.
86
87 f:hash(n) => K constant time.
88 formula, -> K, retrieve using k. O(1)
89 f:max(n) -> linear -> O(n)
90 sorting -> O(nlogn) quick sort, merge sort, n^2
91
92 finding index in list(vector):
93 ["Minu", "Astha", "Shankar", "Ayush", "Saurab"]
94
95 query("Astha") -> 1
96 query("Santa") -> -1
97 O(n)
98 -----
99 sort
100 binary search
101 O(logn)
102 -----
103 HashSet          -> ? unordered. isPresent? O(1)
104 Binary search    -> O(logn)
105 Seq. search      -> O(n)
106
107 n log n
108 -----
109 Fib Series
110 -----
111
112 Give me nth Fib number ?
113 -----
114 N^2
115 NlongN
116 -----
117 normal recursion. n^2
118
119 Memomization
120 -----
121 NlogN
122 ----- You can check it.
123
124 Astha
125 -----
126
127 [a, b, c, a, b, a]
128 trainsform to

```

```

129 [a -> 1, b -> 1, c -> 1, a -> 1, b -> 1, a -> 1]
130
131 group_by
132 first key,
133
134 [a [1, 1, 1], b [1, 1], c [1], d [1]]
135
136 sum value.
137 [a 3, b 2, c 1, d 1]
138
139 Total data N
140 3 N
141 O(N) => linear time.
142
143 -----
144 zip function
145 -----
146 x = [a, b, a, b, a, a]
147 y = [1, 1, 1, 1, 1, 1]
148 -----
149 zip function
150 zip(x,y) => [(a, 1), (b, 1), (a, 1), (b, 1) ...]
151
152 -----
153 [a, b]
154 [0 a, 1 b] => hashmap
155 -----
156 Binod
157 -----
158     1. trying to learn data transformation with other data.
159
160 -----
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192

```

193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215